

## INSTITUTO TECNOLÓGICO DE OAXACA

**Reporte Técnico: Replica de la Aplicación de YouTube para Dispositivos  
Android**

**Integración De Procesos De Desarrollo De Software**

Prof. Espinosa Pérez Jacob

Ángel Arturo Pérez García

28/10/2025

<b>Introducción</b>	<b>1</b>
Objetivo del manual	1
Alcance	1
Público objetivo	1
Resumen	2
<b>Requisitos del sistema</b>	<b>2</b>
Backend	2
Hardware	2
Software	3
Frontend	3
Hardware	3
Software	3
<b>Arquitectura del sistema</b>	<b>4</b>
Diagrama general por módulos del servidor	4
Diagrama general por módulos de la aplicación	5
Diagramas de flujo	5
Videos por geolocalización	5
Búsqueda de videos	5
Registro de usuario	6
Login	6
Selección de video	7
Obtención de Historial	7
Cambio de ubicación	8
Cambio de foto de perfil pt.1	8
Cambio de foto de perfil pt.2	9
Cambio de idioma	9
Cambio de tema	10
Logout	10
Diagramas de secuencia	11
Registro de usuario	11
Login	11
Videos por geolocalización	12
Búsqueda de videos	12
Selección de videos	13
Obtención de historial	13
Cambio de ubicación	14
Cambio de perfil	14
Cambio de idioma	15
Cambio de tema	15

Logout	16
Diagrama ER de la DB	16
<b>Instalación</b>	<b>17</b>
Backend	17
Ubuntu	17
Node.js	20
MySQL Server	21
Frontend	21
Android Studio	21
MySQL	22
Postman	23
<b>Configuración</b>	<b>23</b>
Backend	23
Conexión a la base de datos y creación de usuario	23
Tunnel de Cloudflared para API	23
Frontend	26
Emulador de Android Studio	26
Creación del proyecto	28
<b>Pruebas</b>	<b>29</b>
Backend	29
Frontend	32
<b>Seguridad</b>	<b>36</b>
Backend	36
JWS	36
Middleware de autenticación	37
Restricciones en la credencial de Google Cloud	38
Frontend	38
Token de usuario para peticiones	38
Restricciones en la credencial de Google Cloud	39
<b>Sopporte</b>	<b>39</b>

## Introducción

### Objetivo del manual

El presente Manual Técnico tiene como objetivo proporcionar la información necesaria para instalar, configurar, mantener y comprender la arquitectura interna del sistema de la aplicación móvil desarrollada.

Busca servir como guía para personal técnico, programadores, administradores de sistemas y equipos de soporte, detallando los aspectos estructurales, tecnológicos y de funcionamiento del sistema, con el fin de facilitar futuras modificaciones, actualizaciones o integraciones con otros servicios.

### Alcance

Este manual abarca todos los aspectos técnicos relacionados con el desarrollo y operación de la aplicación Replica de YouTube, una plataforma móvil que utiliza la API de YouTube para mostrar contenido audiovisual geolocalizado.

El documento cubre desde los requerimientos de instalación y configuración del entorno de desarrollo, hasta la implementación de los módulos principales: autenticación, geolocalización, recomendación de videos, historial y consumo de la API externa.

El manual no incluye procesos administrativos, documentación del usuario final ni manuales de operación no técnica, los cuales se presentan en documentos separados.

### Público objetivo

- Desarrolladores de software para aplicaciones con React Native.
- Administradores del sistema backend con Node.js y bases de datos relacionales.
- Estudiantes de carreras relacionadas con la informática.

## **Resumen**

El sistema Replica es una aplicación móvil para SO de Android, desarrollada con tecnologías orientadas a ofrecer una experiencia similar a YouTube, pero con un enfoque geográfico.

Su propósito es permitir que los usuarios descubran y consuman contenido audiovisual generado dentro de su misma zona geográfica, promoviendo la conexión con su comunidad.

La aplicación integra la API de YouTube para obtener datos de videos, canales y estadísticas (visualizaciones, likes, duración, etc.), y utiliza la geolocalización automática o manual del usuario para filtrar el contenido mostrado según su ubicación.

Además, cuenta con módulos de autenticación, historial de búsqueda, sistema de recomendaciones basado en proximidad geográfica, y almacenamiento local de coordenadas para mejorar la experiencia del usuario.

El backend, implementado en Node.js con Express, gestiona la comunicación con la base de datos relacional, MySQL, donde se registran los históricos, referencias de video y usuarios.

Por su parte, el frontend móvil se desarrolló con React Native y el framework de Expo, integrando servicios para la comunicación con el servidor y componentes personalizados para la visualización de videos, mapas y listas de contenido.

## **Requisitos del sistema**

### **Backend**

#### **Hardware**

- Procesador de doble núcleo de 2 GHz
- 4 GB de RAM (memoria del sistema)
- 25 GB de espacio en el disco duro (o memoria USB, tarjeta de memoria o disco externo).
- Una unidad de CD/DVD o un puerto USB para el medio de instalación
- Acceso internet

## **Software**

- Node.js 24.11.0 LTS
- Ubuntu 22.04 LTS
- MySQL Community Server
- Cuenta en Cloudflare con un dominio registrado
- Dependencias:
  - axios: 1.12.2
  - bcryptjs: 3.0.2
  - cookie-parser: 1.4.7
  - cors: 2.8.5
  - dotenv: 17.2.3
  - express: 5.1.0
  - googleapis: 162.0.0
  - jsonwebtoken: 9.0.2
  - mysql2: 3.15.2
  - sequelize: 6.37.7

## **Frontend**

### **Hardware**

- Microsoft Windows 10 de 64 bits
- 16 GB de RAM (memoria del sistema)
- Microarquitectura de CPU posterior a 2017, Intel Core de 8<sup>a</sup> gen. i5 / AMD Zen Ryzen (p. ej., Intel i5-8xxx, Ryzen 1xxx).
- 16 GB de espacio libre
- Resolución de pantalla mínimo de 1280 x 800
- GPU con 4 GB de VRAM, como Nvidia GeForce serie 10 o posterior, o AMD Radeon RX 5000 o posterior con los controladores más recientes

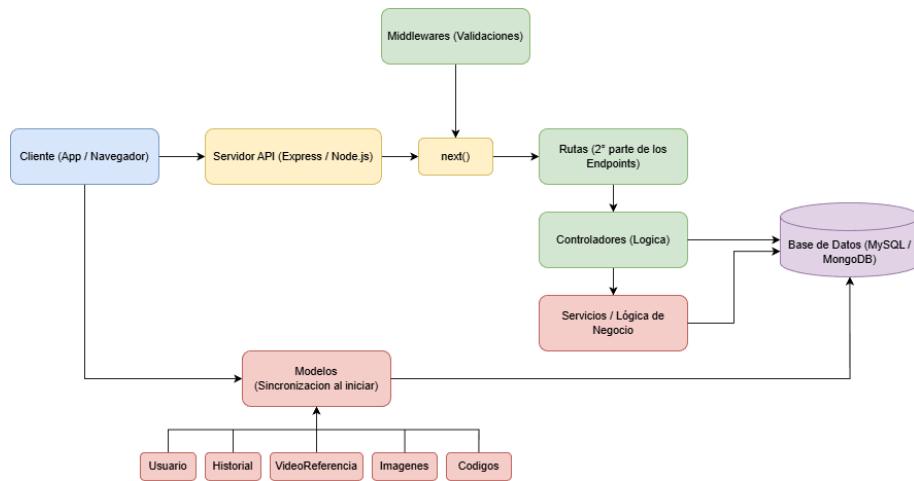
## **Software**

- Android Studio 1.4.8
- Postman
- VSCode (opcional)

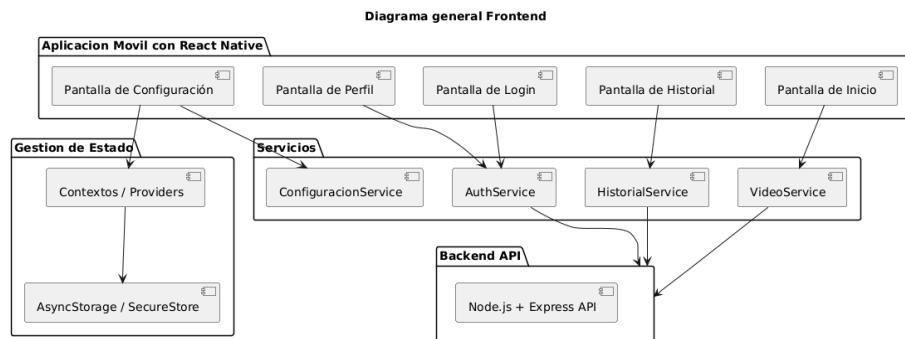
- Dependencias:
  - @expo/vector-icons: ^15.0.3,
  - @react-native-async-storage/async-storage: ^2.2.0,
  - @react-navigation/bottom-tabs: ^7.6.0,
  - @react-navigation/native: ^7.1.19,
  - @react-navigation/stack: ^7.6.0,
  - axios: ^1.13.0,
  - expo: ~54.0.20,
  - expo-auth-session: ~7.0.8,
  - expo-location: ~19.0.7,
  - expo-status-bar: ~3.0.8,
  - expo-web-browser: ~15.0.8,
  - react: 19.1.0,
  - react-native: 0.81.5,
  - react-native-gesture-handler: ~2.28.0,
  - react-native-reanimated: ~4.1.1,
  - react-native-safe-area-context: ~5.6.0,
  - react-native-screens: ~4.16.0,
  - react-native-webview: 13.15.0,
  - react-native-youtube-iframe: ^2.4.1

## Arquitectura del sistema

### Diagrama general por módulos del servidor

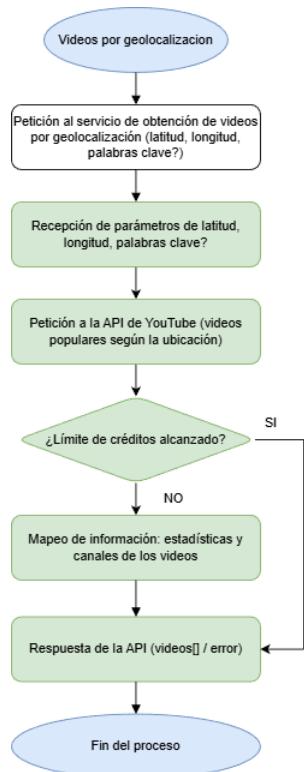


## Diagrama general por módulos de la aplicación

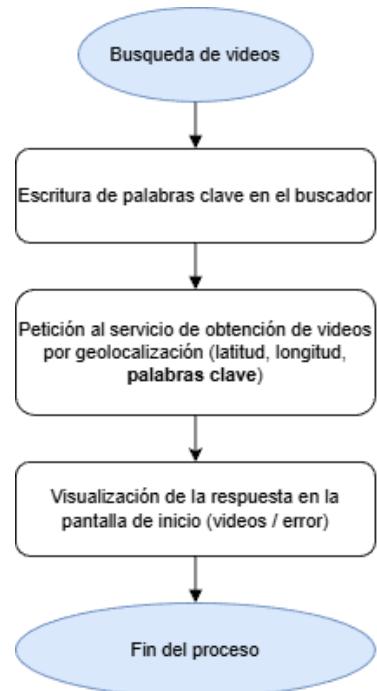


## Diagramas de flujo

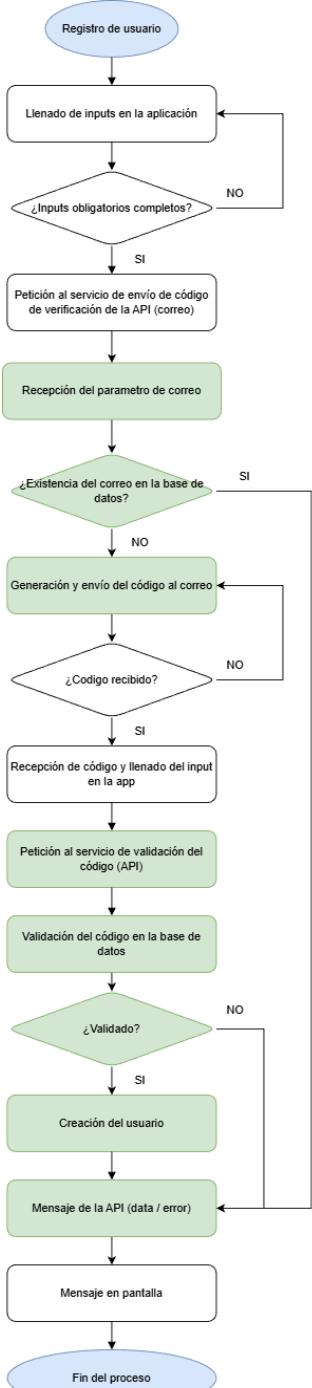
### Videos por geolocalización



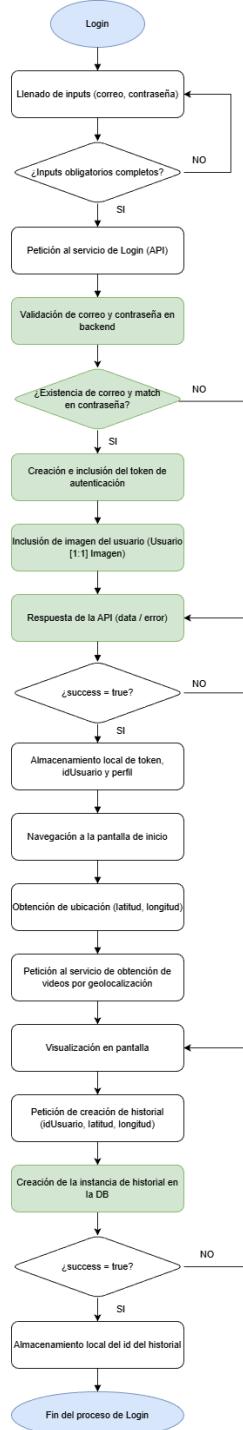
### Búsqueda de videos

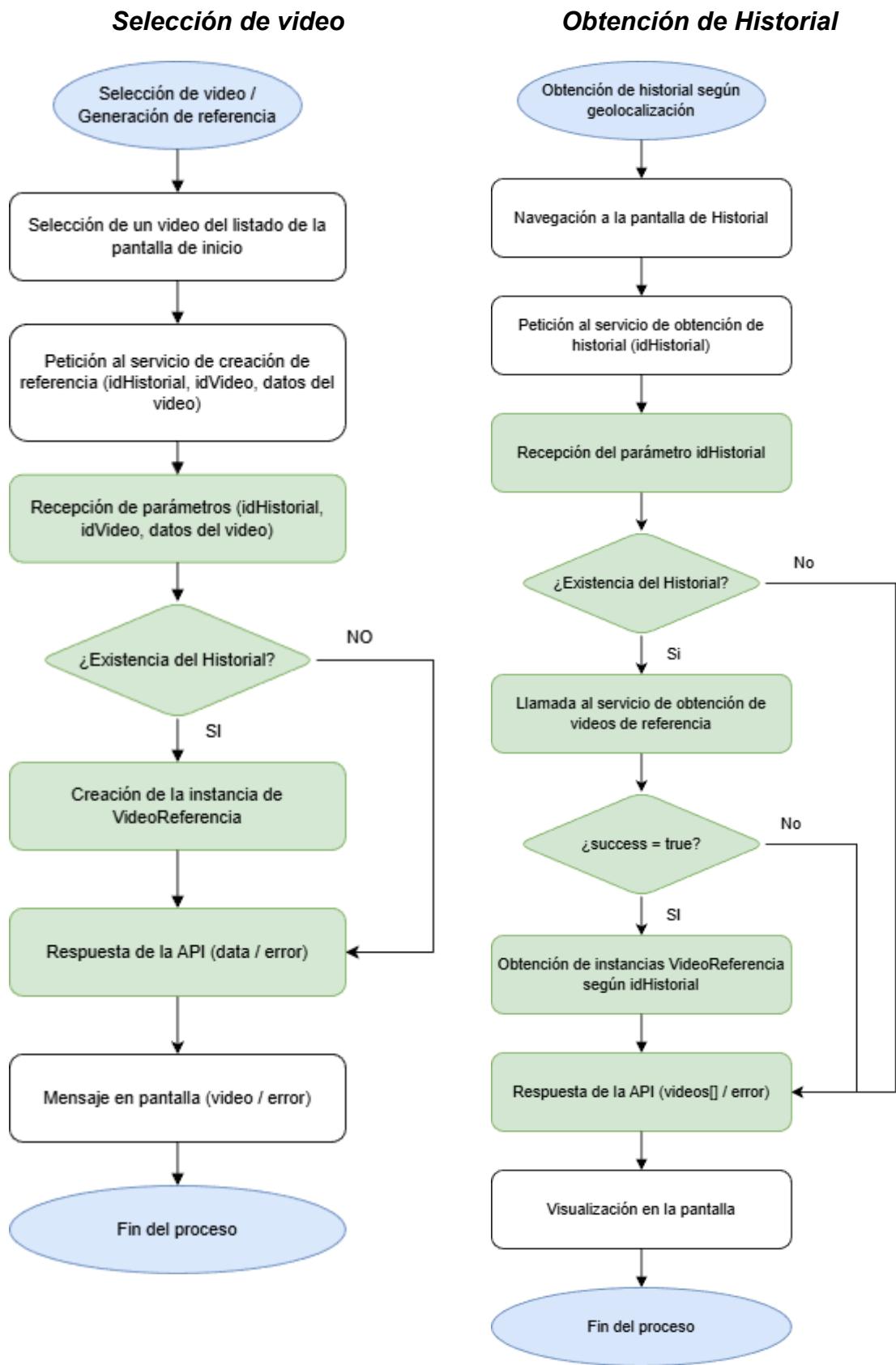


## Registro de usuario

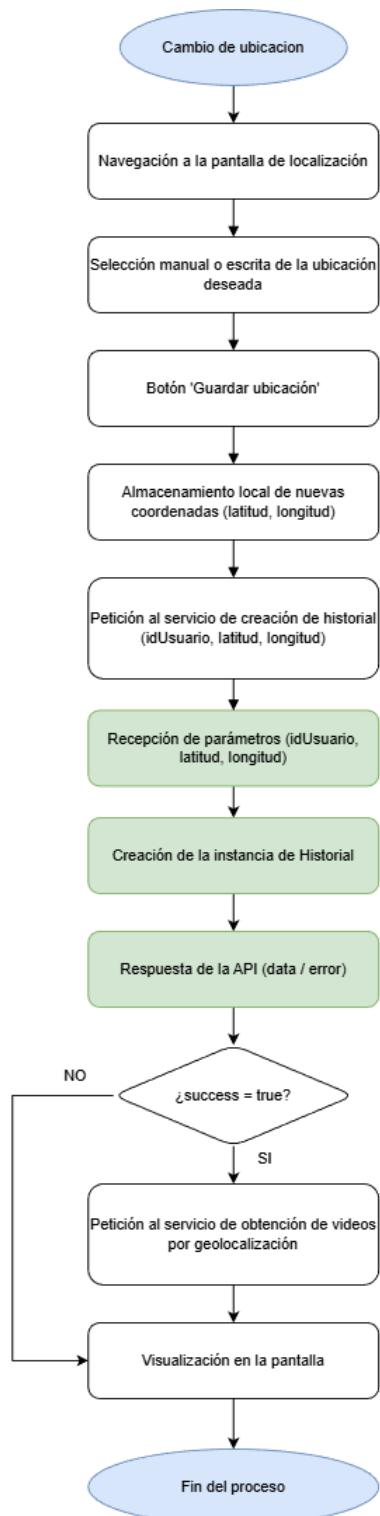


## Login

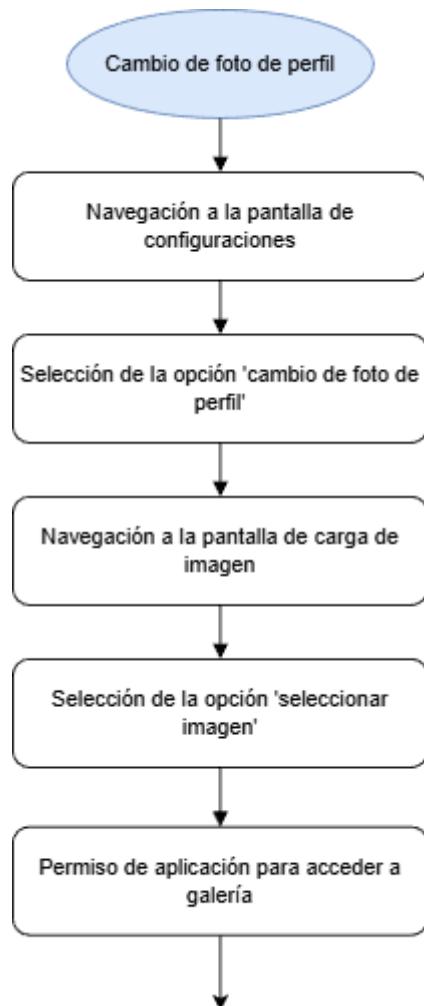




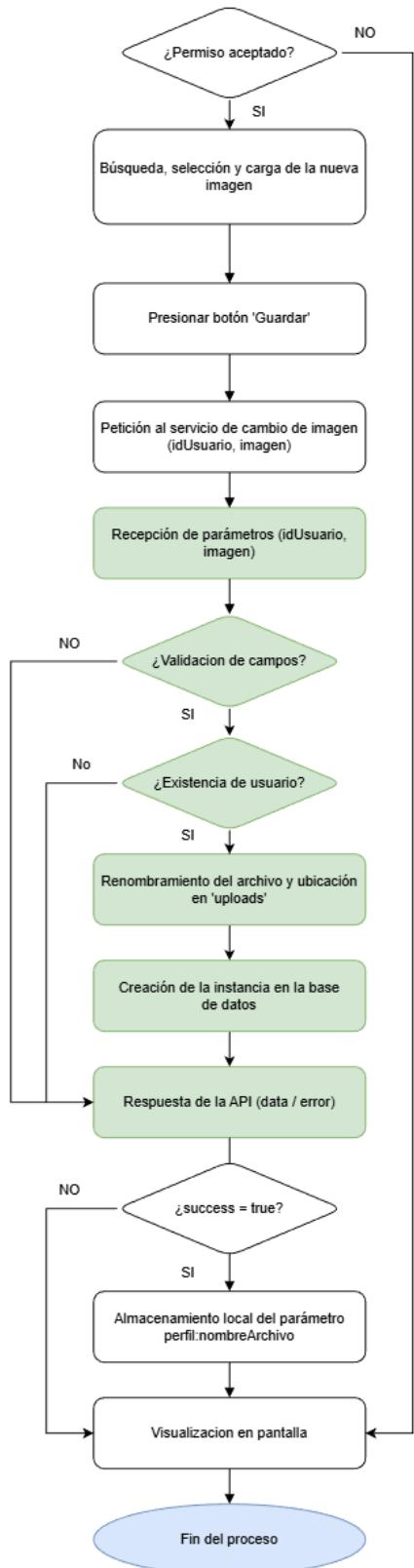
### Cambio de ubicación



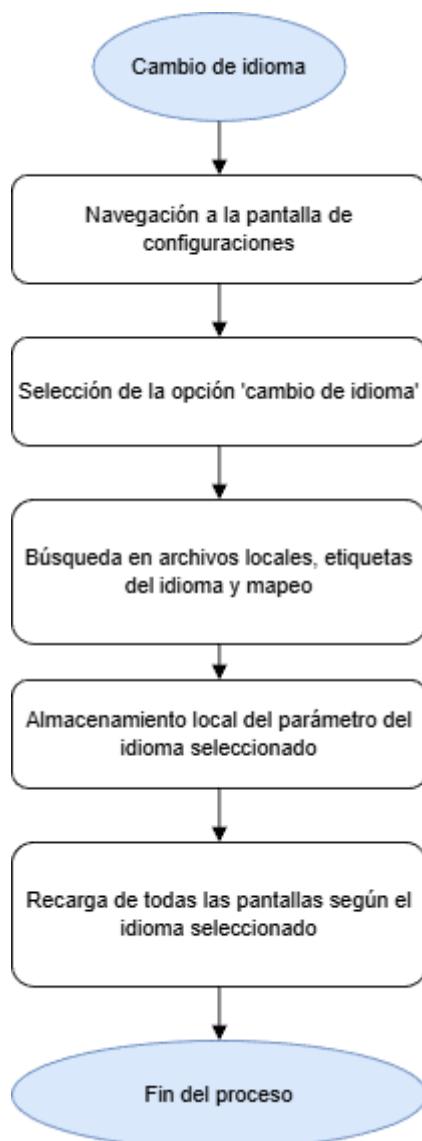
### Cambio de foto de perfil pt.1



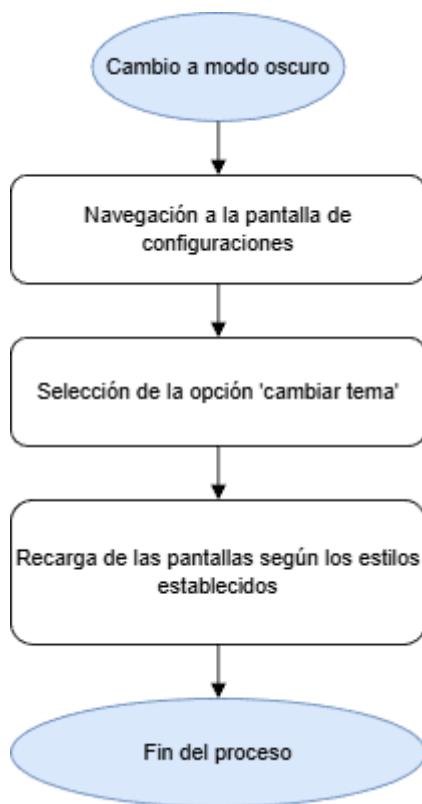
## Cambio de foto de perfil pt.2



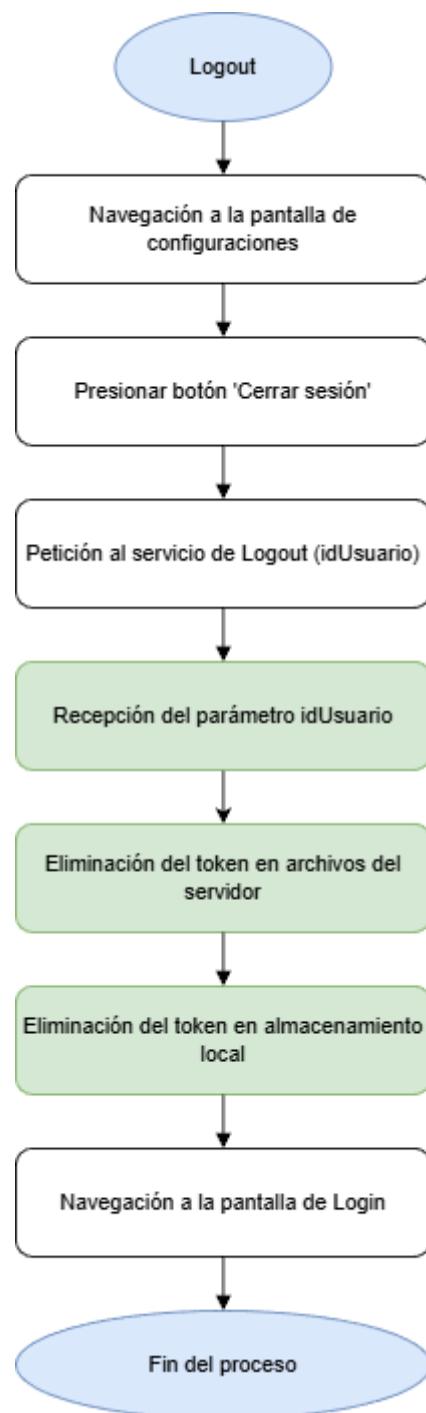
## Cambio de idioma



### **Cambio de tema**

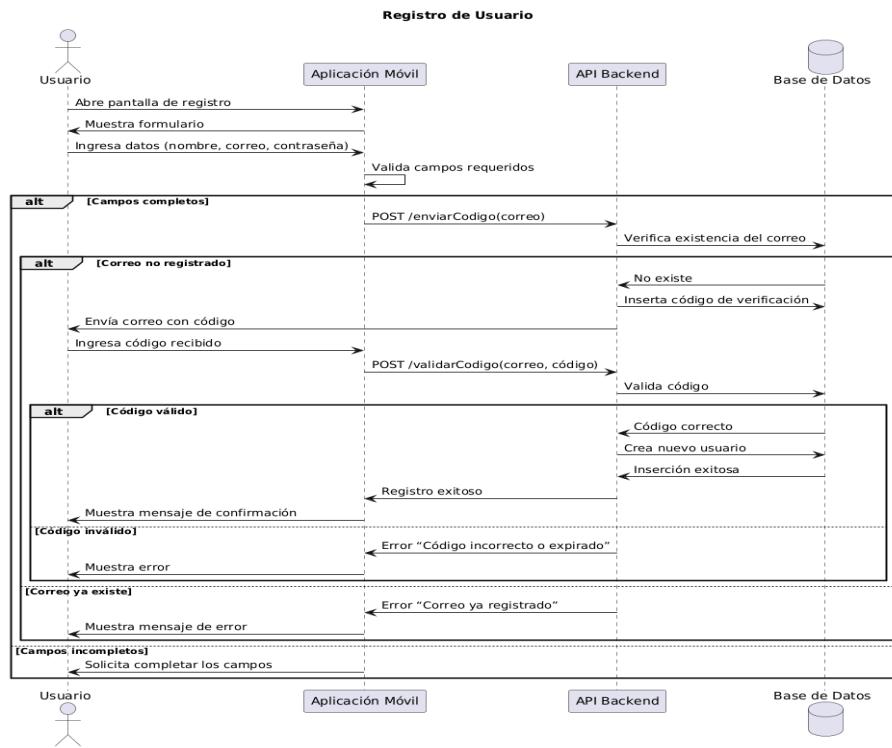


### **Logout**

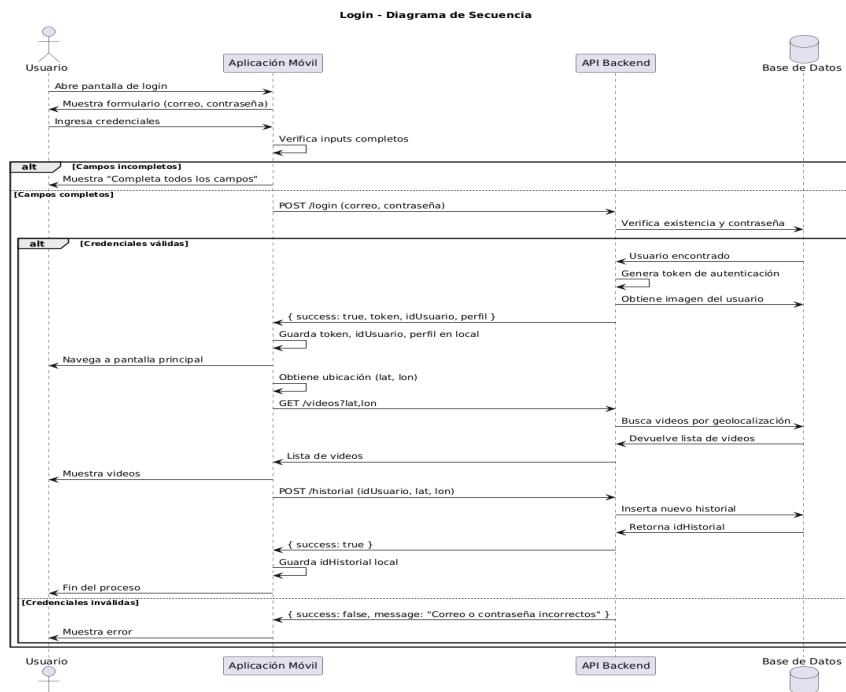


## Diagramas de secuencia

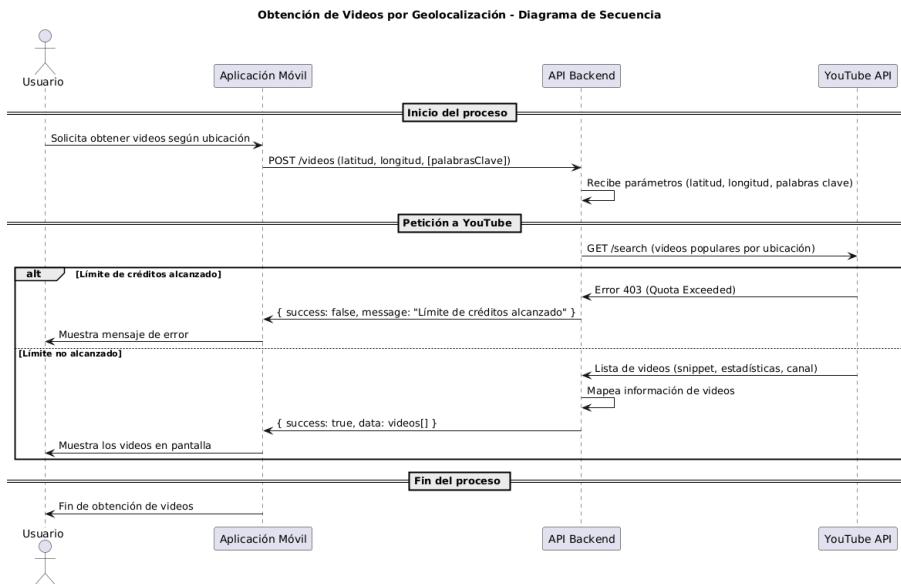
### Registro de usuario



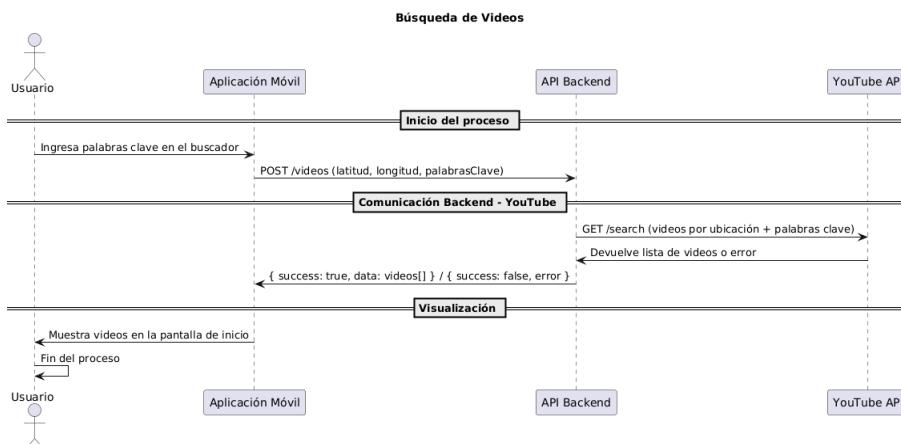
### Login



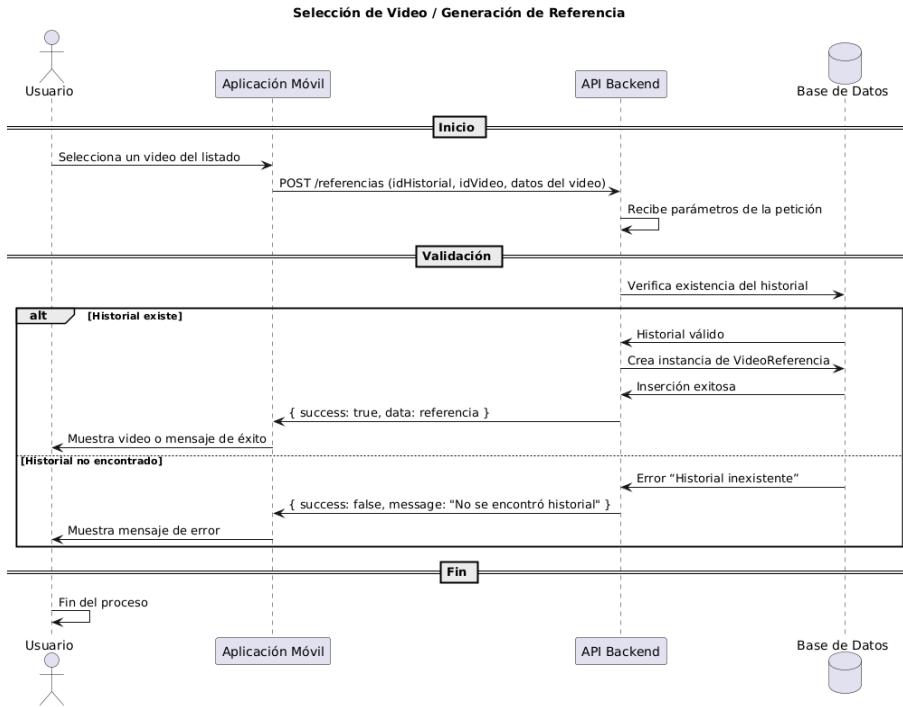
## Videos por geolocalización



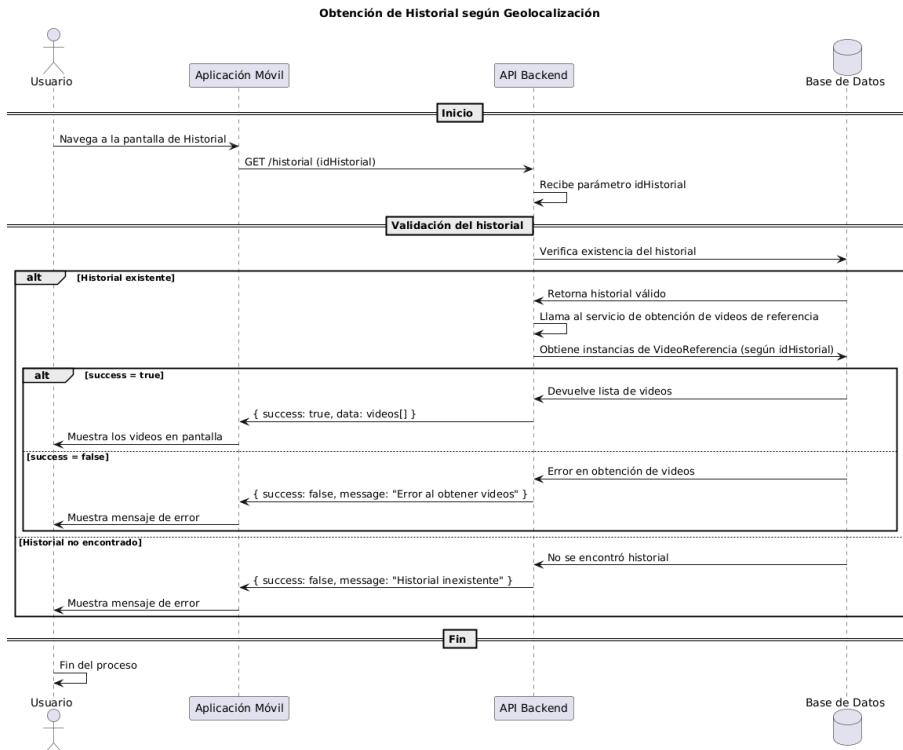
## Búsqueda de videos



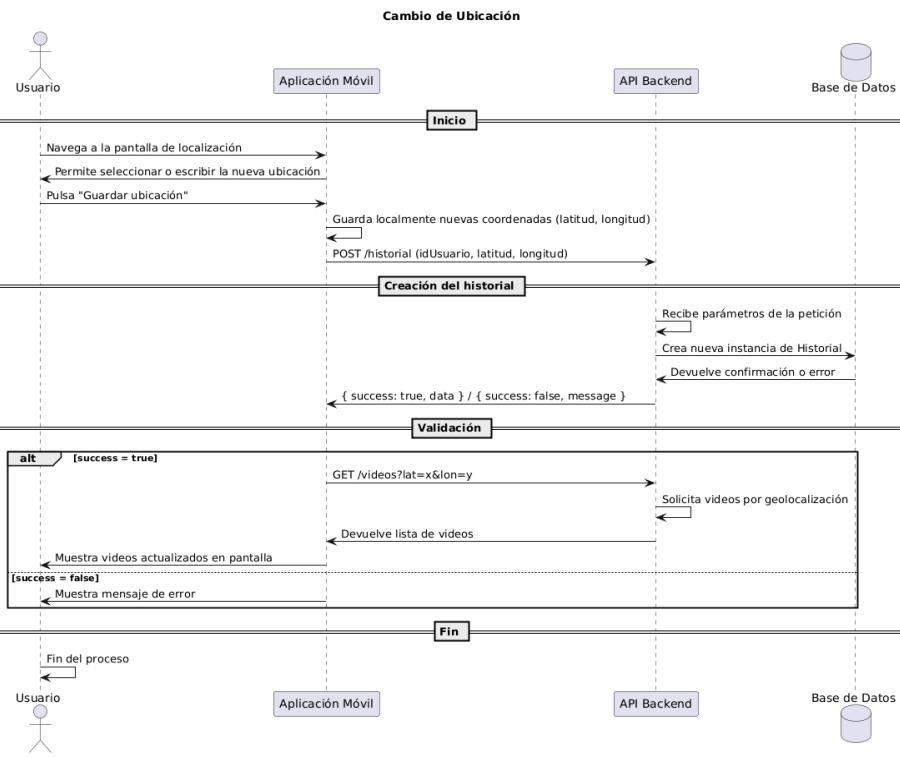
## Selección de videos



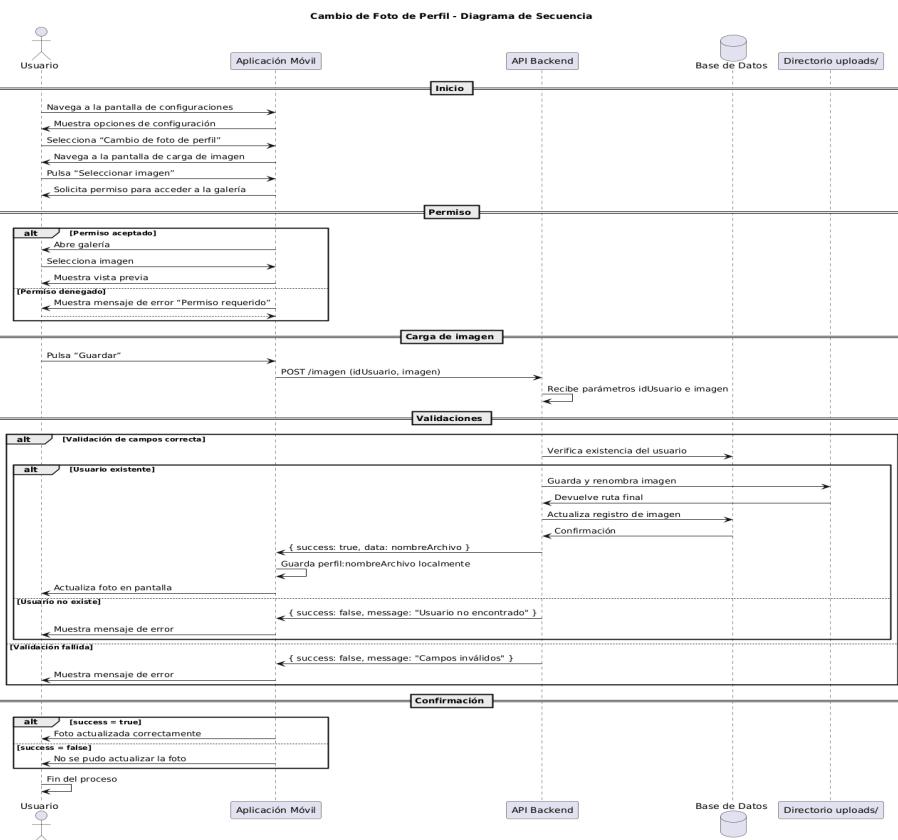
## Obtención de historial



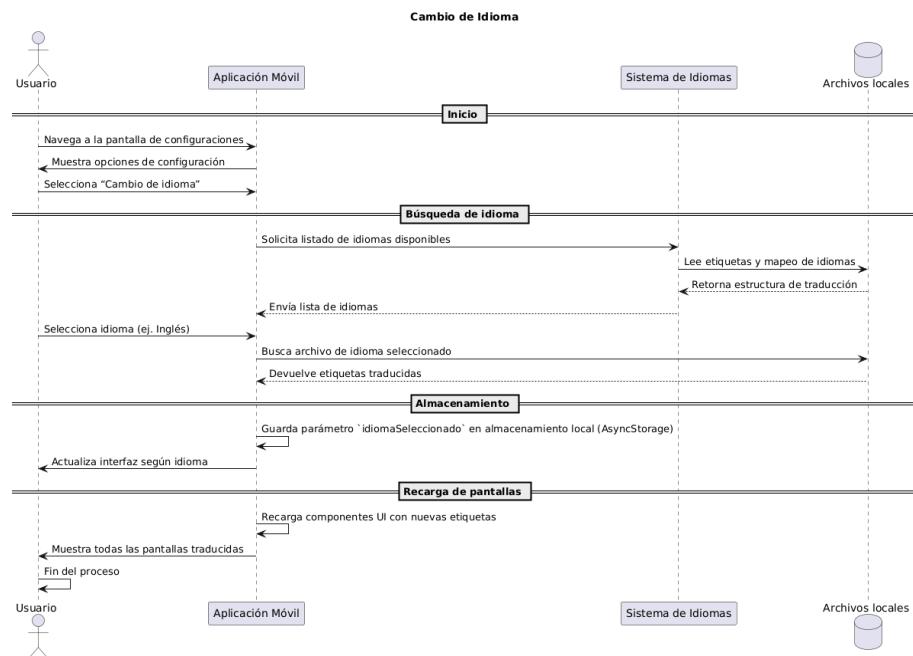
## Cambio de ubicación



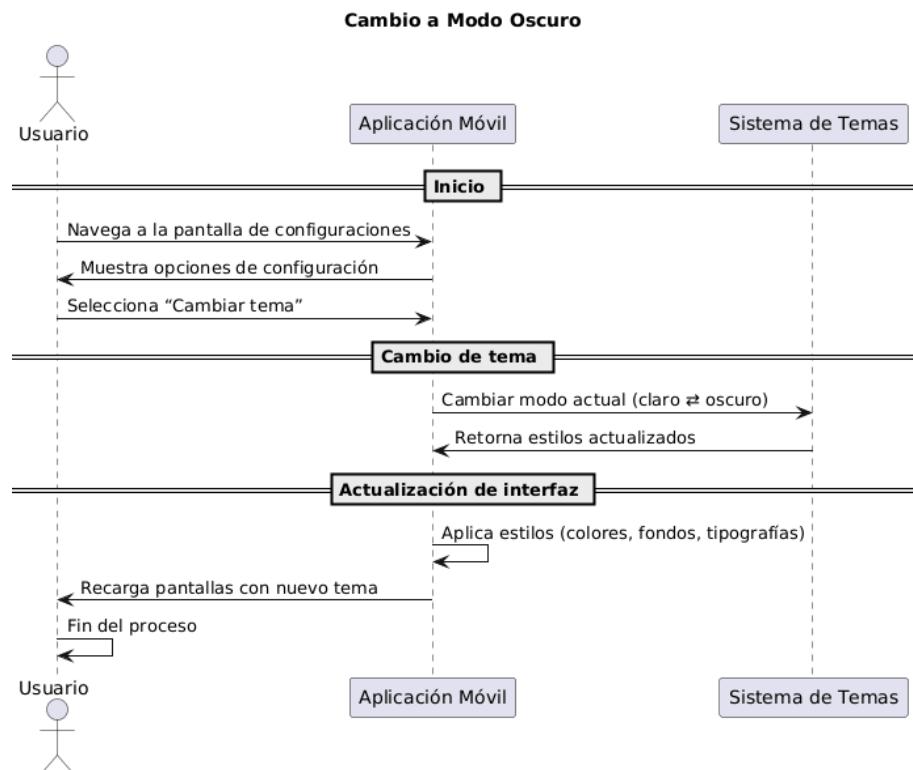
## Cambio de perfil



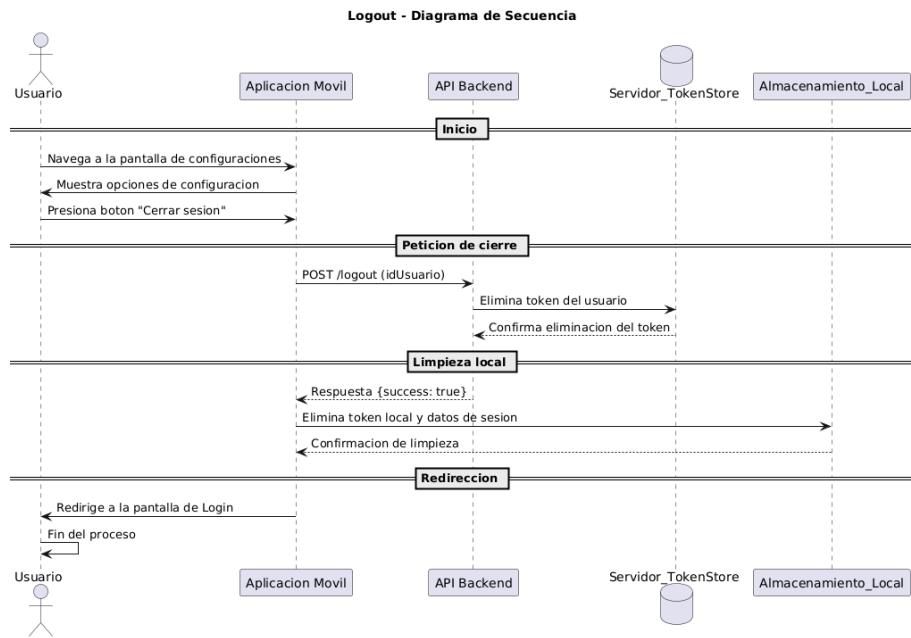
## Cambio de idioma



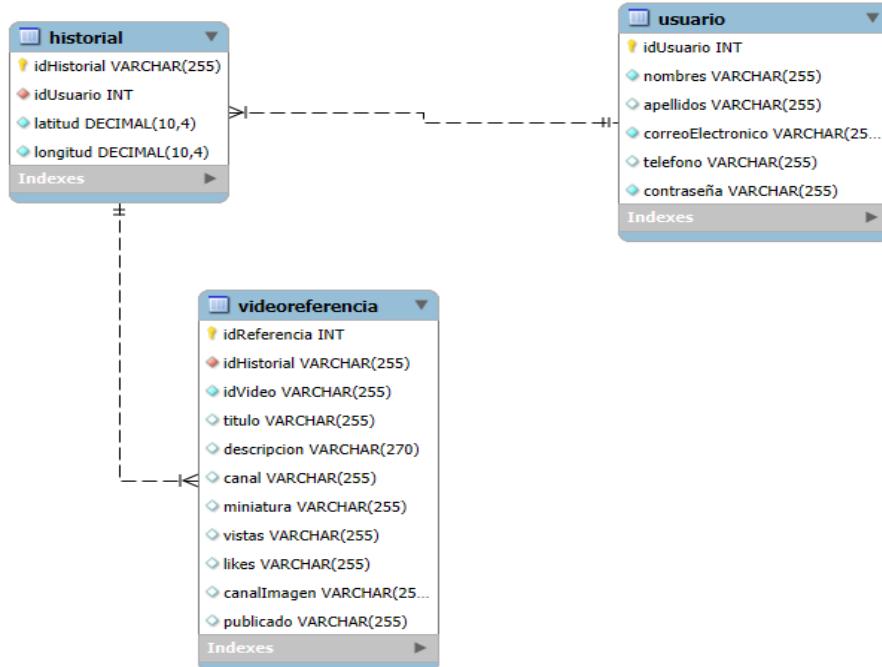
## Cambio de tema



## Logout



## Diagrama ER de la DB



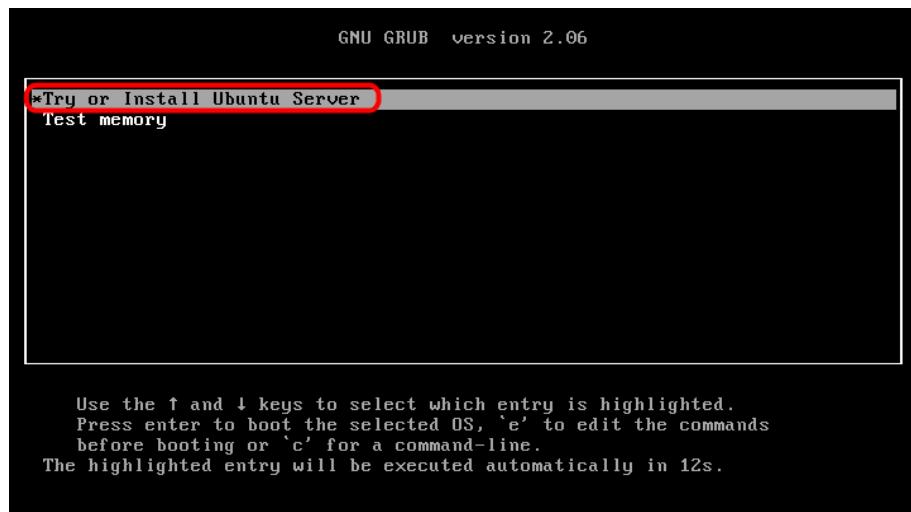
## Instalación

### Backend

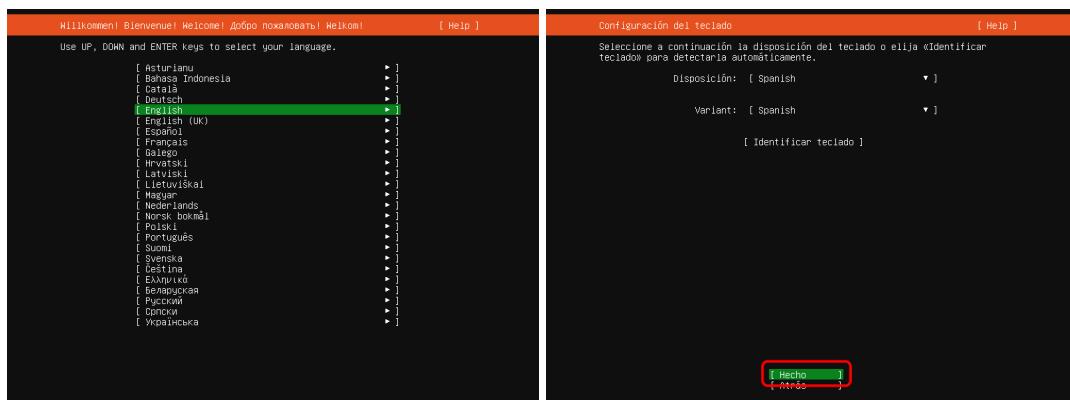
#### *Ubuntu*

Previo a la instalación debemos de tener un instalador USB.

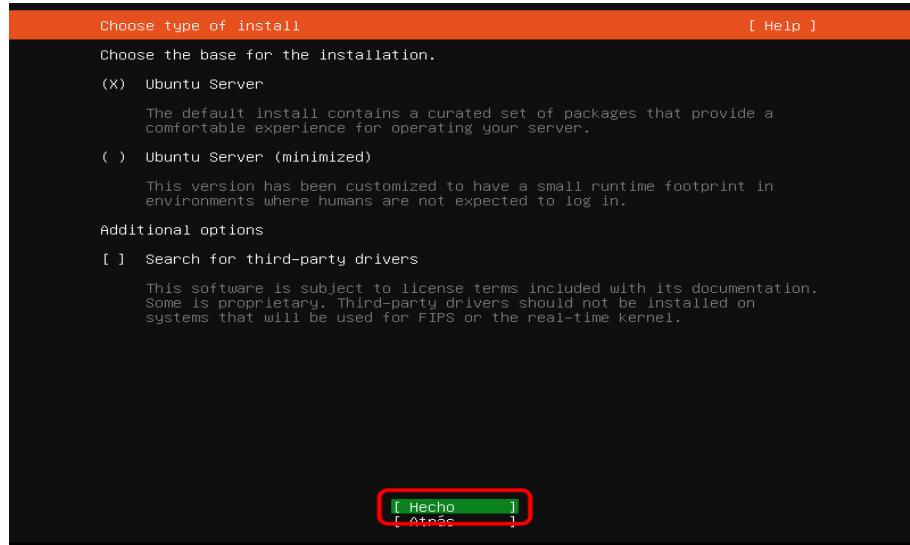
El primer paso es entrar al menú de instalación del sistema, esto puede variar según la máquina que tengamos, en este caso basta con presionar la tecla F9 al momento del arranque. Dentro del menú, si todo salió bien con nuestro instalador, podemos observar habilitada la opción para la instalación del SO.



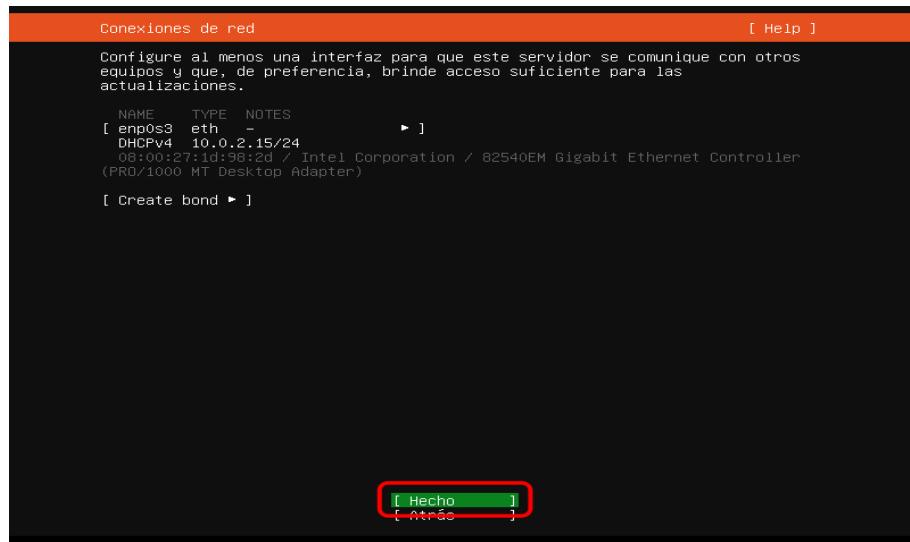
Posteriormente, podemos elegir el idioma del sistema de forma manual, o bien solicitar que el sistema identifique nuestra distribución de teclado y hacer una recomendación de forma automática.



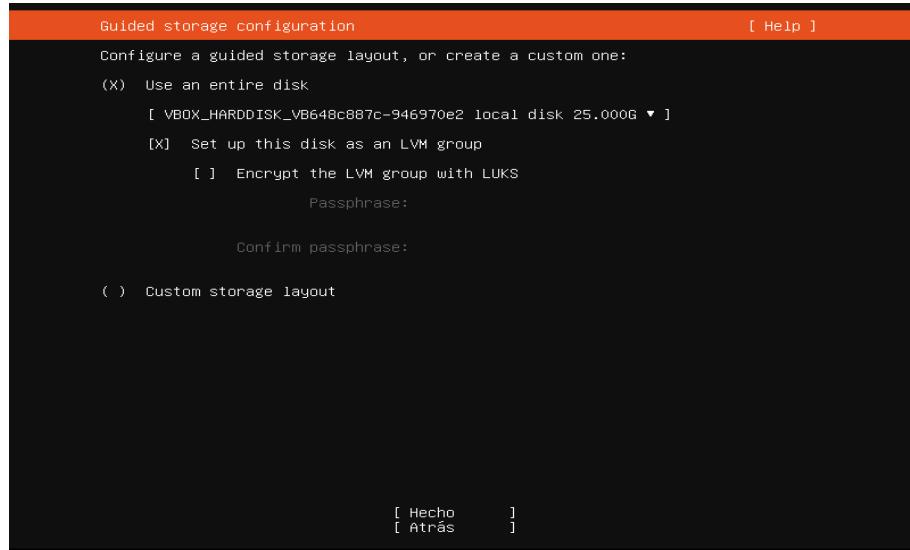
Podremos elegir el tipo de instalación que nos convenga.



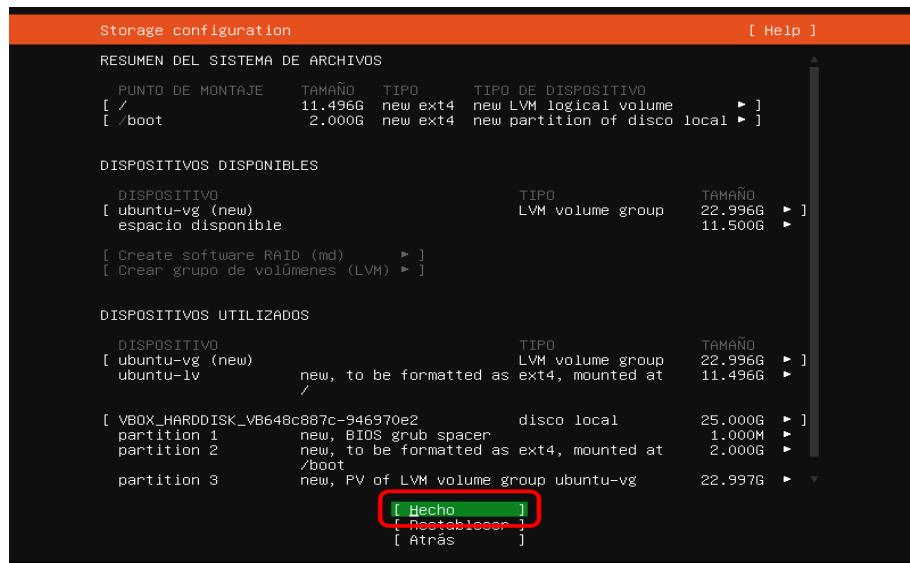
Es recomendable que registremos la red desde la instalación para evitar posibles errores de reconocimiento o configuraciones adicionales en archivos de red, para eso podremos registrar la red en este punto de forma manual, o bien conectar un cable de red para que el sistema lo reconozca automáticamente.



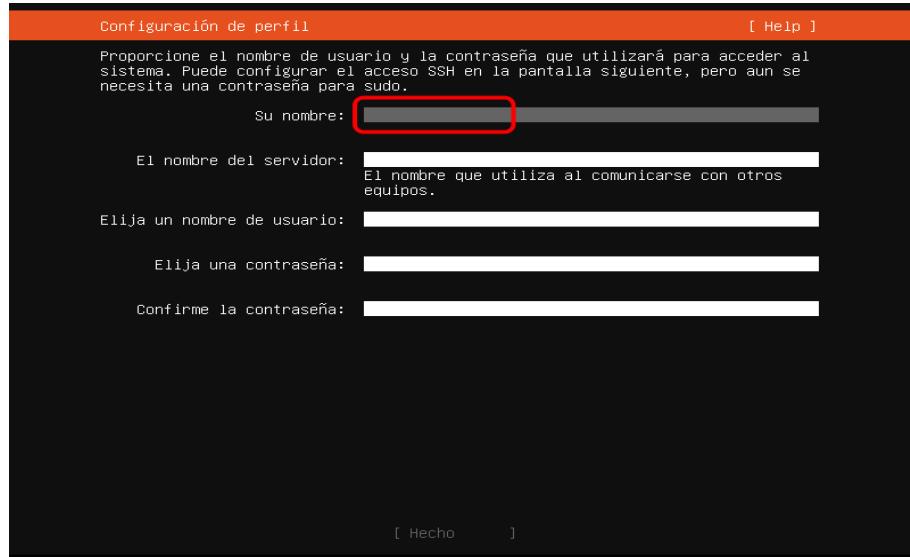
Seleccionamos la partición en la cual alojaremos el SO.



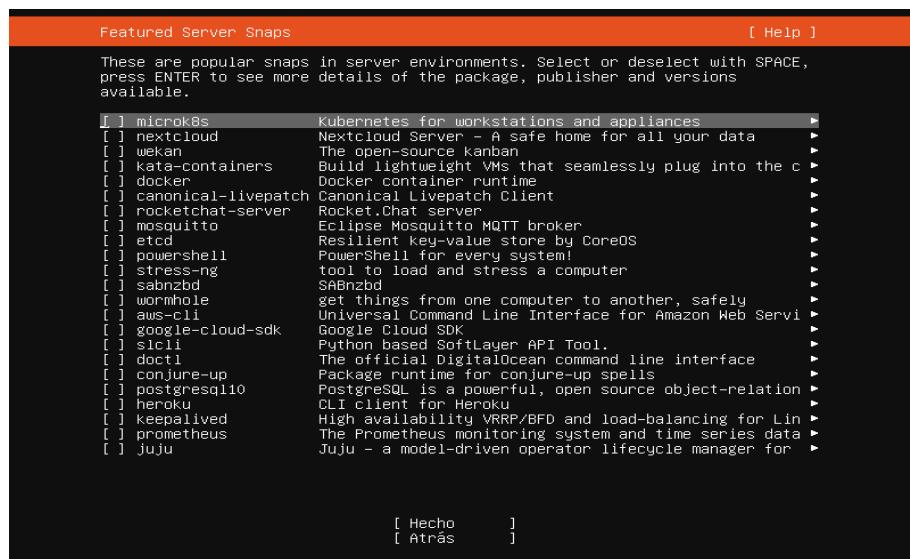
Confirmamos nuestra instalación.



Agregamos las credenciales para nuestro servidor (usuario, contraseña).



Finalmente, esperamos la finalización de la instalación.



## Node.js

Con SSH podemos acceder a nuestro servidor y desde nuestro SO de Windows haremos la instalación de todas las dependencias.

Para la instalación de Node.js usaremos los siguientes comandos.

```
angela@ingreacionprocesos:~$ \_\_ "$HOME/.nvm/nvm.sh"
angela@ingreacionprocesos:~$ nvm install 22
Downloading and installing node v22.21.0...
Downloaded https://nodejs.org/dist/v22.21.0/node-v22.21.0-linux-x64.tar.xz...
#####
Computing checksum with sha256sum
Checksums matched!
Now using node v22.21.0 (npm v10.9.4)
Creating default alias: default -> 22 (-> v22.21.0)
angela@ingreacionprocesos:~$ node -v
v22.21.0
angela@ingreacionprocesos:~$ npm -v
10.9.4
angela@ingreacionprocesos:~$ |
```

## MySQL Server

Para la instalación de MySQL Server haremos uso del siguiente comando:

```
angela@ingreacionprocesos:~$ sudo apt install -y mysql-server mysql-client
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libmysqlclient-dev libmysqlclient20 libmysqlclient20-dev libmysqlclient20-18


```

Posteriormente verificaremos la instalación:

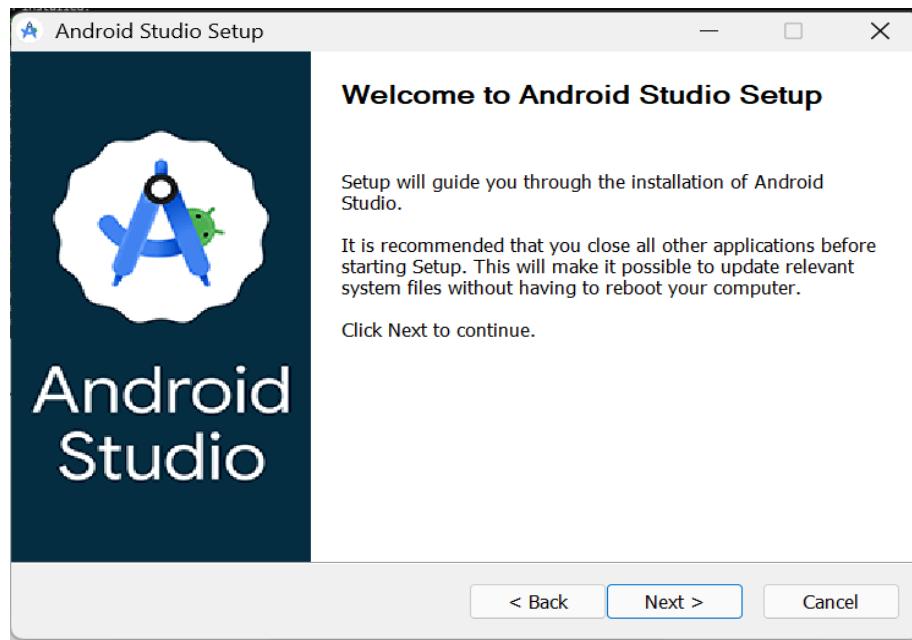
```
angela@ingreacionprocesos:~$ systemctl status mysql.service
● mysql.service - MySQL Community Server
  Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2025-10-21 07:43:09 UTC; 1 week 0 days ago
    Main PID: 960 (mysqld)
      Status: "Server is operational"
        Tasks: 39 (limit: 9282)
       Memory: 506.3M
          CPU: 2h 51min 42.259s
         CGroup: /system.slice/mysql.service
                   └─960 /usr/sbin/mysqld

oct 21 07:42:52 ingreacionprocesos systemd[1]: Starting MySQL Community Server...
oct 21 07:43:09 ingreacionprocesos systemd[1]: Started MySQL Community Server.
angela@ingreacionprocesos:~$ |
```

## Frontend

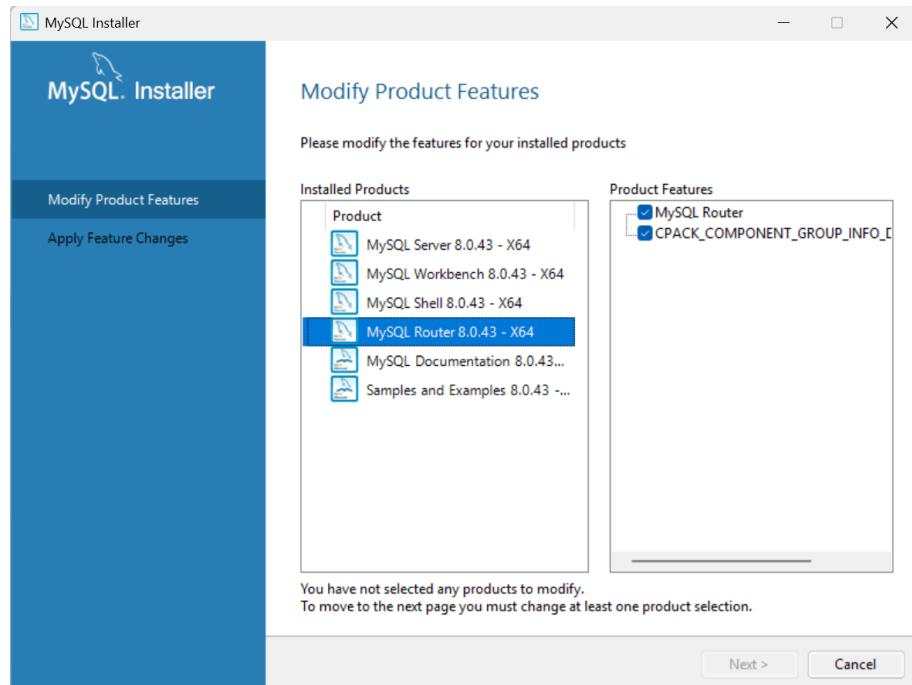
### Android Studio

Para la instalación de Android Studio simplemente presionamos next. La instalación es muy intuitiva.



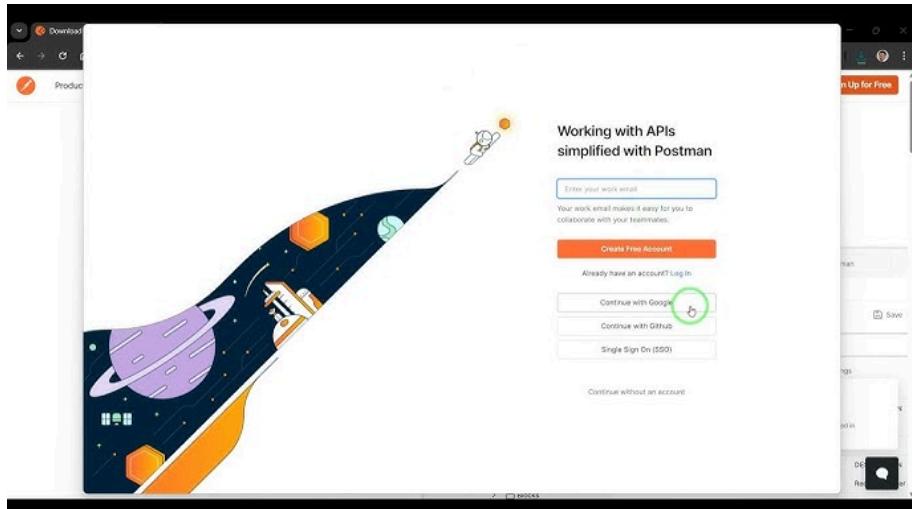
## MySQL

De manera similar que con Android Studio, MySQL tiene una instalación superintuitiva, en este caso hemos hecho la instalación completa del mismo para realizar las pruebas antes del despliegue en el servidor.



## **Postman**

De igual manera, Postman solamente necesita del instalador y una cuenta para poder dar inicio a su uso.



## **Configuración**

## **Backend**

### ***Conexión a la base de datos y creación de usuario***

Una vez instalado el servicio de MySQL debemos de crear un usuario y una base de datos inicial para empezar a trabajar. Para eso ejecutamos los siguientes comandos:

```
create database ejemplo;
```

```
create user 'usuario'@'localhost' identified by 'password_segura';
```

```
grant all privileges on ejemplo.* TO 'usuario'@'localhost';
```

```
flush privileges;
```

```
mysql -u usuario -p
```

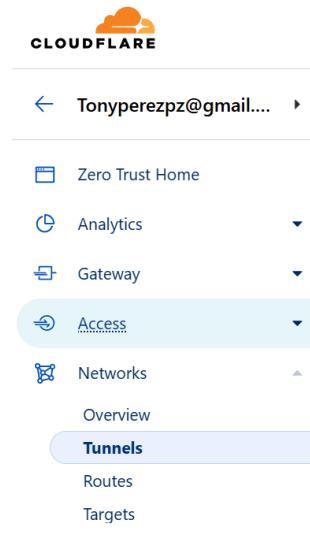
### ***Tunnel de Cloudflared para API***

Lo primero que hay que tener en cuenta para poder hacer una “tunelizacion” de nuestro servidor para exponer la API es contar con una cuenta dentro de la plataforma de Cloudflare y un dominio registrado. Dentro de la misma plataforma podremos comprar uno o bien, si contamos con un dominio externo registrado en

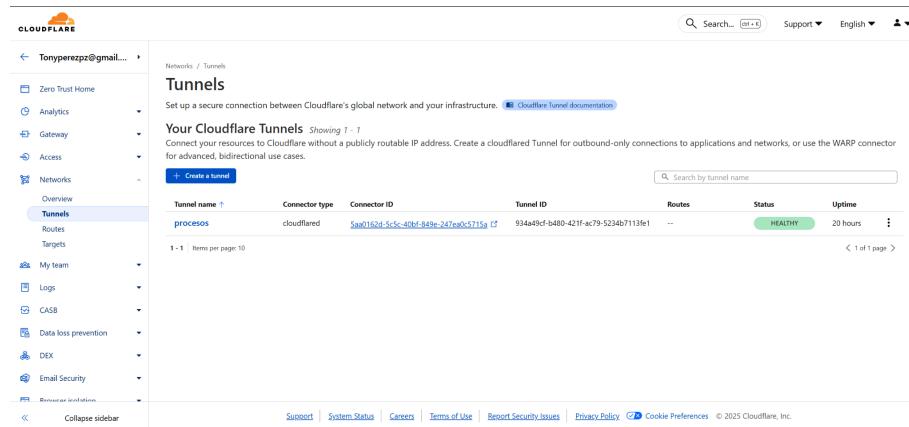
otra plataforma, podremos solicitar el cambio de alojamiento. En este caso, su servidor cuenta con el siguiente “test-drive.org”.

El servicio de túnel que ofrece Cloudflare es gratuito hasta cierto límite. Su política se basa en que hasta que no haya suficiente concurrencia en el dominio, la empresa ofrece el servicio de forma gratuita.

Para poder generar el tunel debemos de ubicarnos dentro de Zero Trust/Networks/Tunnels como se muestra en la imagen.



Dentro encontraremos la opción de “create a tunnel”.



Al seleccionar dicha opción nos podremos elegir entre dos opciones, yo elegiré la opción de “cloudflared”.

Posteriormente, nos pedirá registrar nuestro dominio para vincularlo con el túnel.

Posteriormente, deberemos de elegir las características de nuestro entorno(servidor), en este caso Ubuntu Server es una distro de Linux, por ende seleccionaremos la opción de Debian en 64 bits. Esto nos generará los comandos para poder realizar la configuración de nuestro entorno en el cual queremos que apunte nuestro túnel. Dicha configuración nos dará el apuntador al repo desde él cual se jalará la versión de Cloudflare para su instalación. El comando para guardar el apuntador en nuestro propio repo y el comando de instalación.

Finalmente, nos da el comando para ejecutar nuestro túnel listo para funcionar.

Podremos verificar la instalación con el siguiente comando.

```

angela@ingreacionprocesos:~$ systemctl status cloudflared
● cloudflared.service - cloudflared
   Loaded: loaded (/etc/systemd/system/cloudflared.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2025-10-21 07:43:04 UTC; 1 week 1 day ago
       Main PID: 803 (cloudflared)
          Tasks: 10 (limit: 9282)
            Memory: 58.3M
              CPU: 1h 10min 14.209s
            CGroup: /system.slice/cloudflared.service
                    └─803 /usr/bin/cloudflared run --token eyJhIjoiNjFhZTJkNTUwNDE5MTQ0MjRmZDU1YjFkYzEy

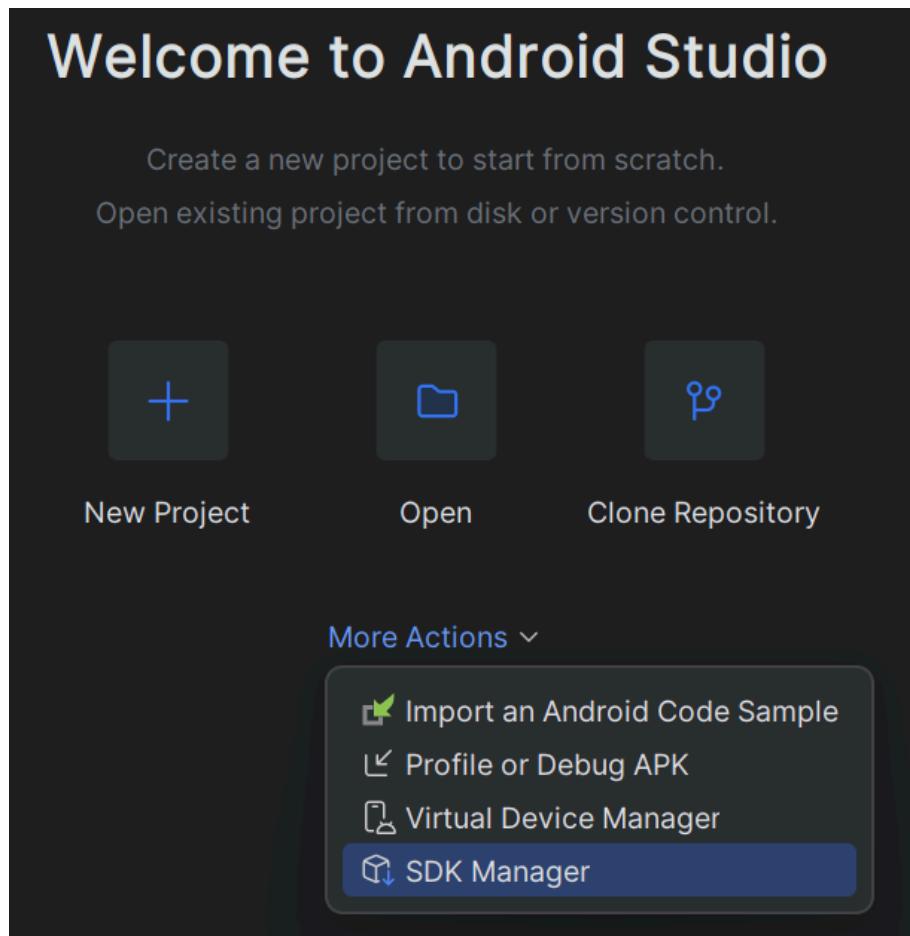
oct 29 05:50:02 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:02Z ERR Connection terminated error="accept stream failed"
oct 29 05:50:02 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:02Z ERR Connection terminated error="datagram manager error"
oct 29 05:50:02 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:02Z ERR Connection terminated error="datagram manager error"
oct 29 05:50:05 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:05Z INF Registered tunnel connection connIndex=0
oct 29 05:50:14 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:14Z INF Tunnel connection curve preferences: [X25]
oct 29 05:50:14 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:14Z INF Tunnel connection curve preferences: [X25]
oct 29 05:50:14 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:14Z INF Tunnel connection curve preferences: [X25]
oct 29 05:50:14 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:14Z INF Registered tunnel connection connIndex=1
oct 29 05:50:14 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:14Z INF Registered tunnel connection connIndex=2
oct 29 05:50:14 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:14Z INF Registered tunnel connection connIndex=3
oct 29 05:50:14 ingreacionprocesos cloudflared[803]: 2025-10-29T05:50:14Z INF Registered tunnel connection connIndex=4
Lines 1-20/20 (END)

```

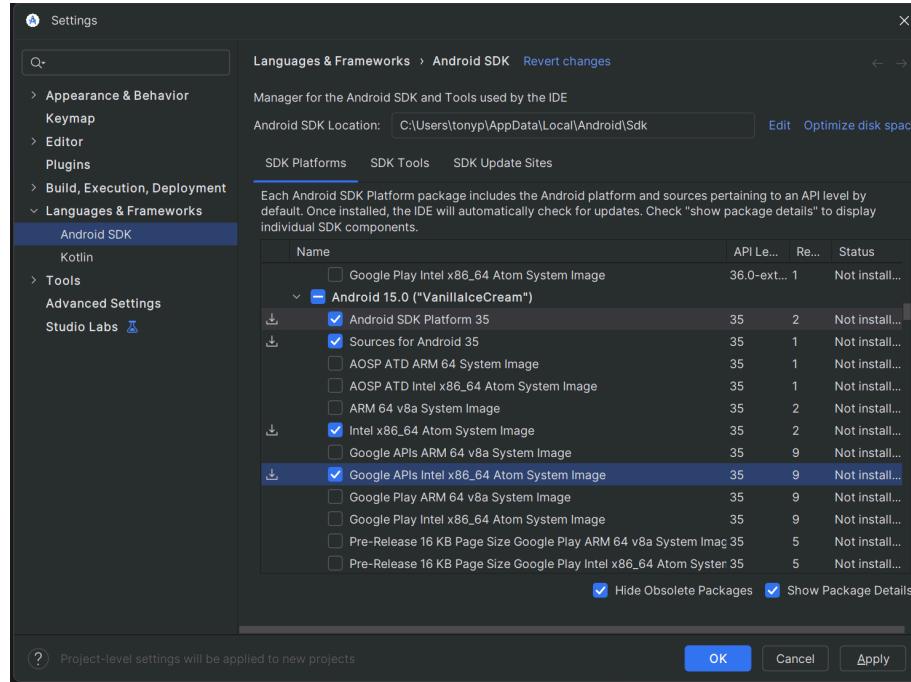
## Frontend

### **Emulador de Android Studio**

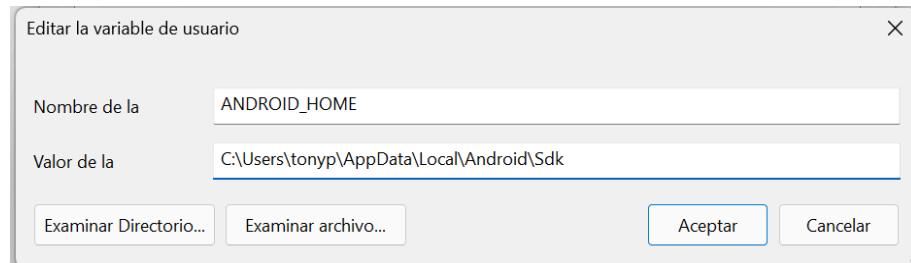
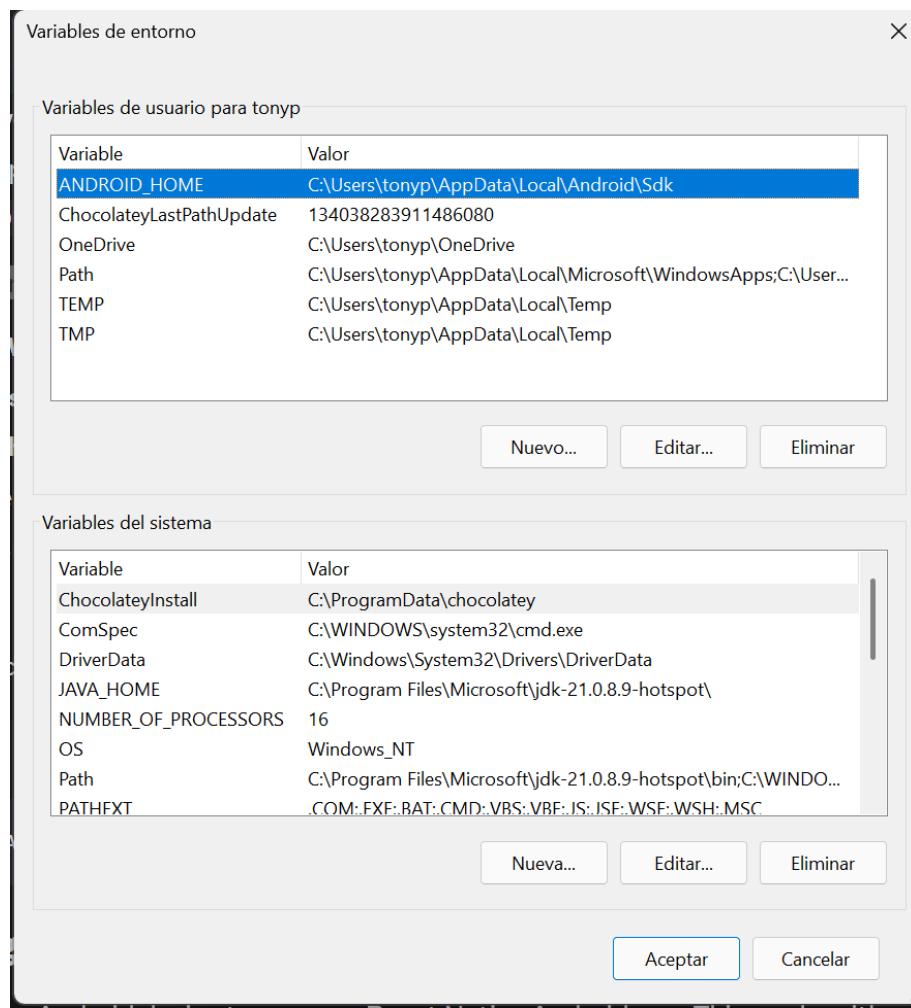
Entramos al programa y buscamos la opción de SDK Manager para realizar la configuración de nuestro emulador.



Buscaremos y seleccionaremos los siguientes paquetes y guardamos los cambios.



Adicionalmente, agregaremos las rutas de las herramientas en las variables de entorno.



## Creación del proyecto

Para dar inicio a un proyecto nuevo con React Native ejecutaremos el siguiente comando:

```
npx create-expo-app@latest Replica –template blank--typescript
```

```

Microsoft Windows [Versión: 10.0_26100_6589]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\tonyp>cd C:\Users\tonyp\OneDrive\Documentos\Expo_projects
C:\Users\tonyp\OneDrive\Documentos\Expo_projects>npx create-expo-app@latest replica --template blank-typescript
Need to install the following packages:
  create-expo-app@3.5.3
OK to proceed? (y)
Creating an Expo project using the blank-typescript template.
// Downloaded and extracted project files.

npm install
  npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
  npm warn deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
  npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
  npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
  npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
  npm warn deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
added 731 packages, and audited 732 packages in 55s
60 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
Your project is ready!

To run your project, navigate to the directory and run one of the following npm commands.
- cd replica
- npm run android
- npm run ios # you need to use macOS to build the iOS project - use the Expo app if you need to do iOS development without a Mac
- npm run web

C:\Users\tonyp\OneDrive\Documentos\Expo_projects>

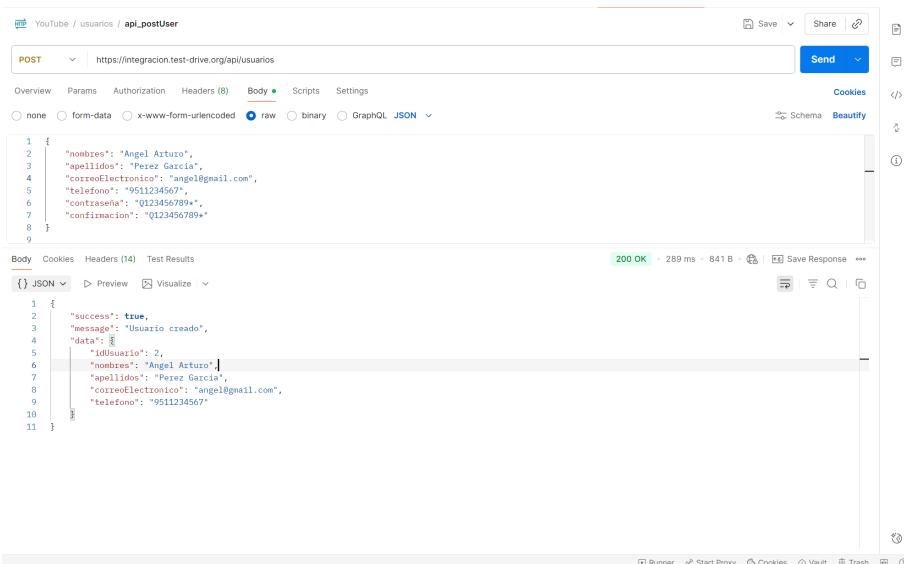
```

## Pruebas

### Backend

#### Llamada a la ruta de la API del proyecto para la creación de una instancia de Usuario

Los parámetros básicos para la creación de la instancia son “correoElectronico” y “contraseña”, los demás parámetros son opcionales.



#### Llamada a la ruta de la API del proyecto para la creación de una instancia de Historial

La ruta de la API recibe tres parámetros, el “idUsuario”, “latitud” y “longitud”. Esto para asegurar que cada instancia de los usuarios en función a su locación sea única.

```

POST https://integracion.test-drive.org/api/historiales
{
  "idUsuario": "1",
  "latitud": "13.9031",
  "longitud": "15.9031"
}

```

```

{
  "success": true,
  "message": "Registro de historial creado",
  "data": {
    "idHistorial": "1_13.9031_15.9031",
    "idUsuario": "1",
    "latitud": "13.9031",
    "longitud": "15.9031"
  }
}

```

## Llamada a la ruta de la API del proyecto para la creación de una instancia de VideoReferencia

La instancia de VideoReferencia debe estar asociada a una instancia de Historial, a su vez el idVideo el cual se renderizará en al app.

```

POST https://integracion.test-drive.org/api/referencias
{
  "idHistorial": "1_13.9031_15.9031",
  "idVideo": "X8S6x8JtJ_Q"
}

```

```

{
  "success": true,
  "message": "Video agregado correctamente al historial.",
  "data": [
    {
      "idReferencia": 1,
      "idHistorial": "1_13.9031_15.9031",
      "idVideo": "X8S6x8JtJ_Q"
    }
  ]
}

```

## Llamada a la ruta de la API del proyecto para el Logueo de cuenta

Para hacer uso posterior de las rutas de API es necesario contar con una cuenta dentro de la app, por ende las llamadas posteriores a los servicios serán verificadas a través de un token de autorización, el cual se entregara al usuario al Loguearse.

```

POST https://integracion.test-drive.org/api/login
{
    "correoElectronico": "angel@gmail.com",
    "contraseña": "Q123456789"
}
200 OK
{
    "success": true,
    "mensaje": "Autenticado correctamente",
    "token": "eyJhbGciOiJIUzI1NiIsInBpdjI6IkpXVCJ9.eyJpc3MiOiJsb2dpbi5jb250ZWdlcjIiLCJleHAiOjE3NjE4NTkyMz19.1K_kyCoxv9BYV9yHtRnuXtpo8zDmrt2wz7XtYqEi10",
    "usuario": {
        "idUsuario": 2,
        "nombre": "Angel Arturo",
        "correoElectronico": "angel@gmail.com"
    }
}

```

## Llamada al servicio de la API del proyecto para el consumo de la API de YouTube

La API ya cuenta con la credencial para realizar las llamadas a la API de YouTube dentro de sus variables de entorno, la siguiente ruta se encarga de verificar la cuenta del usuario para dentro de la app y redirigir la petición.

```

GET https://integracion.test-drive.org/api/youtube/populares?lat=19.4326&lon=-99.1332
200 OK
{
    "success": true,
    "data": [
        {
            "id": "IXJ5nj9ePKQ",
            "titulo": "LOGRE EL JUEGO DEL CALAMAR 🎯 #shorts @SoyTioNestor",
            "descripcion": "",
            "canal": "LORETTA",
            "minatura": "https://i.ytimg.com/vi/IXJ5nj9ePKQ/hqdefault.jpg",
            "vistas": "300376",
            "fecha": "2023-10-27T00:00:00Z",
            "duracion": "PT54S",
            "canalImagen": "https://yt3.ggpht.com/JieIPDRHpkG8Ad_1114KUfievlX6MzJ0D27_ih8868q_b3uNMxvnK3zXMud-Wehw7a0syBu_k=s88-c-k-c0x0fffff-no-r1",
            "ubicacion": [
                {
                    "lat": "19.4326",
                    "lon": "-99.1332",
                    "radio": "10km"
                }
            ],
            "publicado": "2023-10-27"
        }
    ]
}

```

## Llamada al servicio de la API del proyecto para la obtención de las instancias de VideoReferencia asociadas a un Historial

Dado que las instancias de VideoReferencia están asociadas a un Historial, la siguiente ruta se encarga de “jalar” todas las referencias de un Historial y devolverlas en un arreglo para la futura vista de Historial de la app.

The screenshot shows a Postman interface with the following details:

- URL:** https://integracion.test-drive.org/api/historiales/113.9031\_15.9031/referencias
- Method:** GET
- Headers:** (7) - Authorization, Content-Type, User-Agent, etc.
- Query Params:** (1) - Key: Description, Value: Description
- Body:** (1) - JSON (Pretty Print)
- Test Results:** 200 OK, 126 ms, 923 B
- Response Body (JSON):**

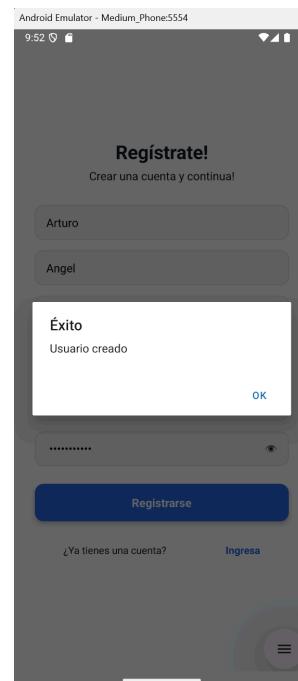
```

1 {
2   "success": true,
3   "message": "Referencias obtenidas correctamente.",
4   "data": [
5     {
6       "idReferencia": 1,
7       "idHistorial": "113.9031_15.9031",
8       "idVideo": "X8S6x8jTJ_Q",
9       "titulo": null,
10      "descripcion": null,
11      "canal": null,
12      "miniaturna": null,
13      "tags": null,
14      "likes": null,
15      "canalImagen": null,
16      "publicado": null
17    },
18    {
19      "idReferencia": 2,
20      "idHistorial": "113.9031_15.9031",
21      "idVideo": "X8S6x8jTJ_Q"
22    }
  ]
}

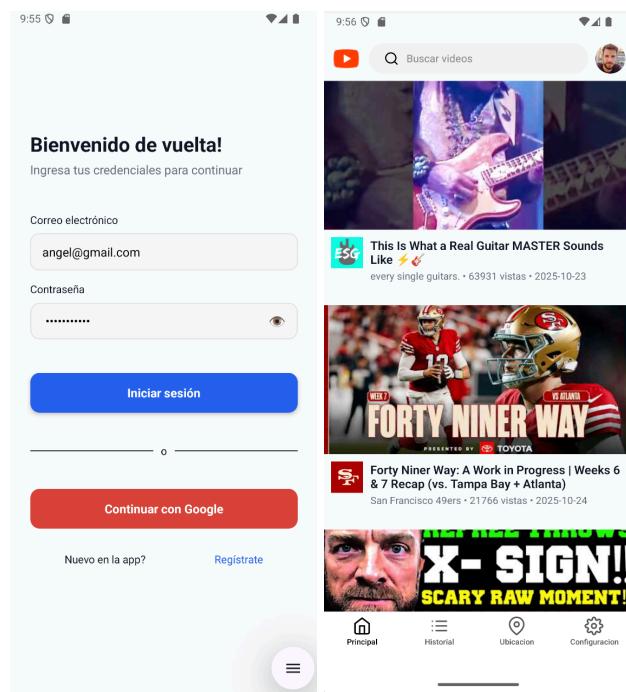
```

## Frontend

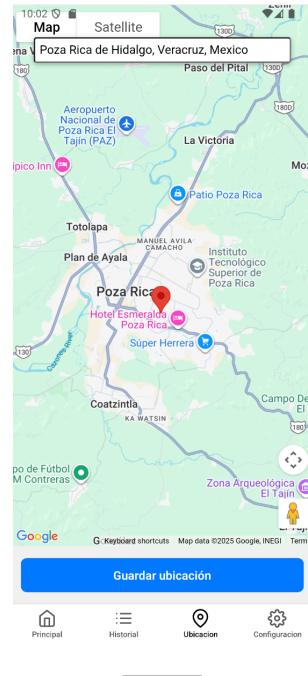
### Registro de usuario



## Logueo de cuenta



## Selección de locación



## Cambio de videos en función a la locación



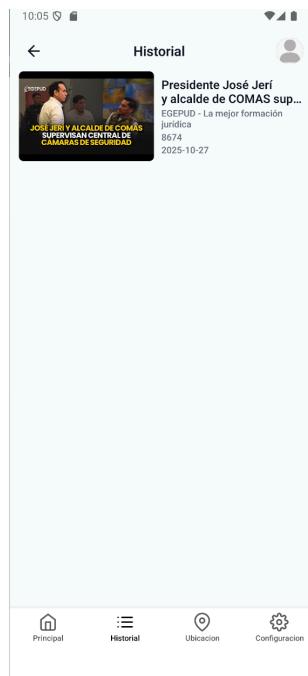
## Búsqueda personalizada de videos en función a palabras clave



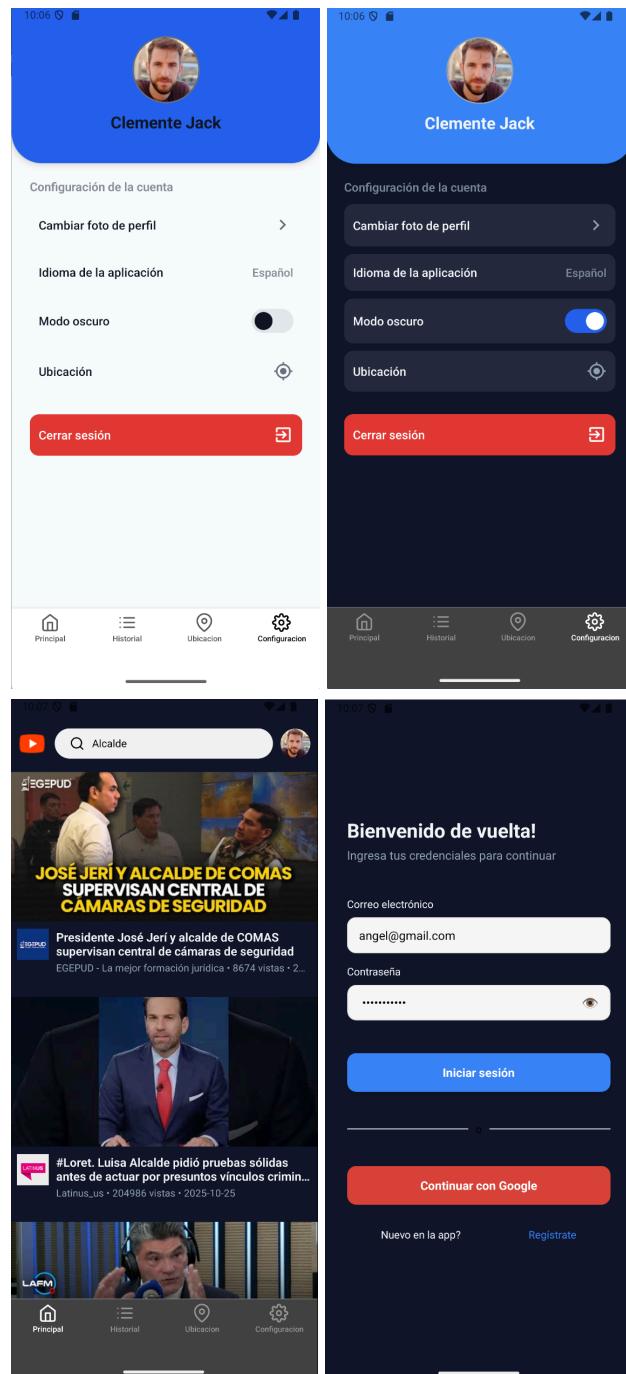
## Selección de videos



## Historial de videos seleccionados



## Cambio de tema de la app



## Seguridad

### Backend

#### JWS

Dentro de nuestro controlador de autenticación contaremos con un método JWS para generar y firmar el token de respuesta del servidor.

```
// Generar token JWT
const token = jwt.sign(
  {
    idUsuario: usuario.idUsuario,
    correoElectronico: usuario.correoElectronico,
  },
  TOKEN_SECRET,
  { expiresIn: "1d" }
);
```

### **Middleware de autenticación**

La llamada a las rutas del servidor antes de servir los recursos pasan a través del middleware de autenticación, este se encarga de recibir el token de autenticación que genera el servidor y que recibe el usuario, y validar que dicho token fue generado por el servidor, si es así entonces continuará con todo el proceso.

```
// endpoints
app.use("/api", logRoute);
app.use("/api/usuarios", userRoute);
app.use("/api/historiales", validarToken, recordRoute);
app.use("/api/referencias", validarToken, videoReferenceRoute);
app.use("/api/youtube", validarToken, youtubeRoutes);
```

```
// Si no existe token en ningún lado
if (!token) {
  return res.status(401).json({
    success: false,
    message: "Acceso denegado: no se proporcionó token",
  });
}

// Verificar token
const verificado = jwt.verify(token, TOKEN_SECRET);

// Guardar los datos decodificados (id, correo, etc.)
req.user = verificado;

next(); // continuar hacia el controlador
```

## ***Restricciones en la credencial de Google Cloud***

La clave del backend para el consumo de las APIs de Google tendrá dos restricciones: de dirección y a los servicios a los cuales podrá realizar las peticiones, como se muestra en la imagen.

The screenshot shows the 'Restrictions' section of a Google Cloud API key. It includes:

- A radio button group for application types:
  - Ninguno (unselected)
  - Sitios web (unselected)
  - Direcciones IP (selected)
  - Apps para Android (unselected)
  - Apps para iOS (unselected)
- A heading 'Restricciones de direcciones IP' with a note: 'Especifica una o más direcciones IP de los emisores que tienen permiso para usar tu clave de API. Aplica el formato de una dirección IPv4 o IPv6, o de una subred con notación CIDR. Ejemplos: 192.168.0.1, 172.16.0.0/12, 2001:db8::1 o 2001:db8::/64'. Below it is a table with a single row:  Dirección IP (checkbox), 177.238.21.64 (text), and an 'Editar' (Edit) button with a pencil icon.
- A heading 'Restricciones de API' with a radio button group:
  - No restringir clave (unselected)
  - Esta clave puede llamar a cualquier API (text)
  - Restringir clave (selected)
- A dropdown menu showing '3 API'.
- A section titled 'API seleccionadas:' listing three APIs: YouTube Data API v3, Maps JavaScript API, and Geocoding API.

## **Frontend**

### ***Token de usuario para peticiones***

El token resultante del login de la aplicación se almacena en los archivos locales para mayor practicidad.

```

const login = async (email: string, password: string) => {
  try {
    setLoading(true);
    const data = await loginRequest(email, password);

    setToken(data.token);
    await AsyncStorage.setItem("token", data.token); // almacenamiento local del token (AsyncStorage)
    // al hacer login se guarda el idUsuario para futuras peticiones
    await AsyncStorage.setItem("idUsuario", data.usuario.idUsuario.toString());
    // Configura el token para futuras peticiones HTTP
    axios.defaults.headers.common["Authorization"] = `Bearer ${data.token}`; // <-----
  } catch (error: any) {
    // console.error("Error de login:", error.response?.data || error.message);
    throw new Error("Credenciales inválidas o error de conexión");
  } finally {
    setLoading(false);
  }
};

```

## **Restricciones en la credencial de Google Cloud**

La única restricción para la credencial del frontend será en los servicios a los cuales podrá realizar las peticiones

The screenshot shows the 'Restrictions' section of a Google Cloud API key. It includes fields for 'Key restrictions' (set to 'None'), 'Application restrictions' (set to 'None'), and 'API restrictions' (set to 'Restrict key'). A tooltip for application restrictions explains they limit API usage to specific apps or devices. Below these are sections for 'Selected APIs' (Maps JavaScript API and Places API) and 'Supported' (with a link to support.google.com).

Restricciones de clave

Fecha de creac

Restricciones de clave

Agrega restricciones para reducir el riesgo de seguridad y evitar el uso no autorizado. [Más información](#)

Restricciones de aplicaciones

Ninguno

Sitios web

Direcciones IP

Apps para Android

Apps para iOS

Las restricciones de aplicaciones limitan el uso de las claves de API a sitios web, direcciones IP y aplicaciones para Android o iOS específicas. Puedes configurar una restricción de aplicaciones por clave.

Restricciones de API

No restringir clave

Esta clave puede llamar a cualquier API

Restringir clave

2 API

API seleccionadas:

Maps JavaScript API

Places API

**Soporte**

Cualquier duda en relación con el proyecto pueden escribirme en mi correo personal.

**4n6el.4rtur0@gmail.com**